UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/671,889	09/29/2003	Fred Gehrung Gustavson	YOR920030170US1	8009

48150 7590 04/07/2009
MCGINN INTELLECTUAL PROPERTY LAW GROUP, PLLC
8321 OLD COURTHOUSE ROAD
SUITE 200
VIENNA, VA 22182-3817

EXAMINER	
VICARY, KEITH E	

ART UNIT	PAPER NUMBER
2183	

MAIL DATE	DELIVERY MODE
04/07/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/671,889	GUSTAVSON ET AL.	
	Examiner	**Art Unit**	
	Keith Vicary	2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE _3_ MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on _21 January 2009_.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) _1-9 and 11-19_ is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) _1-9 and 11-19_ is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-9 and 11-19 are pending in this office action and presented for examination. Claims 1-2, 6, 12-13, and 17 are newly amended by amendment filed 8/20/2008.

Specification

2. The amendment filed 1/21/2009 is objected to under 35 U.S.C. 132(a) because it introduces new matter into the disclosure. 35 U.S.C. 132(a) states that no amendment shall introduce new matter into the disclosure of the invention. The added material which is not supported by the original disclosure is as follows.

Applicant is required to cancel the new matter in the reply to this Office Action.

3. The second paragraph added to the specification discloses that k > 1 indicates a number of data capable of being simultaneously moved in a single instruction, which is of a different scope than the '888 application's apparent teaching that k > 1 indicates a machine has multiple SIMD FPUs.

Additionally, it is noted that a recitation which originates in a co-pending application which discloses of SIMD considerations would not necessarily support an SIMD-based recitation in the instant claims *in the context of the instant claims*. In other words, a disclosure of an invention in the context of an SIMD architecture in a co-pending application would not necessarily support claims directed to the *instant invention* in the context of an SIMD architecture. Therefore, if the SIMD/k>1 aspect is expanded upon in a future set of claims, applicant should explicitly describe how the

instant invention in particular, in view of the co-pending applications, supports that claimed subject matter.

4. Applicant has added into the fourth paragraph the recitation "[h]owever, this layout can be proved to be one-dimensional". However, this recitation does not appear to be in the original disclosure or explicitly in the other co-pending applications. If this recitation is indeed supported by the original disclosure or the other co-pending applications, examiner recommends explicitly citing to the examiner the location of this recitation.

Claim Rejections - 35 USC § 112

5. The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

6. Claims 1-9 and 11-19 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention.

7. Claim 1 recites the limitation “LSUs can load said data into said Fregs before it is scheduled to be used in said linear algebra subroutine execution” in lines 8-10. The original disclosure does not disclose the broad interpretation of the claim in which the data is loaded into said Fregs before the instruction which uses said data is scheduled for execution. An amendment to overcome the corresponding indefinite rejection in a manner consistent with the original specification would most likely overcome this rejection.

Examiner recommends rewording the claim to eliminate the “scheduling” aspect, as a time at which an instruction which uses data is *scheduled* for later execution is different from a time when an instruction which uses data is actually executed.

8. Claim 1 recites the limitation “a register block format predetermined to reduce a number of data streams for a level 3 nested loop matrix-matrix type kernel type operation processing (e.g., level 3 processing) to be three streams” in lines 10-12. This limitation does not appear to be present in the original disclosure of the instant application. If the limitation is present somewhere in one of the co-pending applications, this should be noted in any subsequent arguments to overcome the rejection. Applicant has previously argued that the present application supports the claim language of reducing the number of data streams to be three streams via various citations. However, the citations given do not appear to support the claim language of *reducing* the number of data streams to be three streams.

Examiner recommends rewording the claim to eliminate the “reducing” aspect, or giving a specific citation which explicitly notes that a reduction is occurring. It is noted that although a reduction may be occurring in view of past prior art implementations, the original disclosure nevertheless must disclose of this reduction in order for it to be valid inside the claim.

a. Claims 2-5 are rejected for inheriting the defects of base claim 1.

9. Claim 6 recites the limitation “said three data streams comprise data of one matrix…and data for two remaining matrix operands…” in the last 4 lines. The original disclosure does not disclose the broad interpretation of the claim in which each data stream contains data of all three matrixes. An amendment to overcome the corresponding indefinite rejection in a manner consistent with the original specification would most likely overcome this rejection as well.

Examiner notes this issue has for the most part been corrected in the other independent claims.

10. Claim 6 recite the limitation “a format predetermined to reduce a number of data streams for a level 3 linear algebra processing to be three streams” in lines 10-11. This limitation does not appear to be present in the original disclosure of the instant application. If the limitation is present somewhere in one of the co-pending applications, this should be noted in any subsequent arguments to overcome the rejection. Applicant has previously argued that the present application supports the claim language of reducing the number of data streams to be three streams via various citations.

However, the citations given do not appear to support the claim language of *reducing* the number of data streams to be three streams.

Examiner recommends rewording the claim to eliminate the “reducing” aspect, or giving a specific citation which explicitly notes that a reduction is occurring. It is noted that although a reduction may be occurring in view of past prior art implementations, the original disclosure nevertheless must disclose of this reduction in order for it to be valid inside the claim.

b. Claims 7-9 and 11 are rejected for inheriting the defects of base claim 6.

11. Claim 12 recites the limitation “inserting instructions to move data into said cache providing said data into said FPU before it was scheduled to be used for processing in said linear algebra subroutine” in lines 8-10. The original disclosure does not disclose the broad interpretation of the claim in which the data is loaded into said Fregs before the instruction which uses said data is scheduled for execution. An amendment to overcome the corresponding indefinite rejection in a manner consistent with the original specification would most likely overcome this rejection as well.

Examiner recommends rewording the claim to eliminate the “scheduling” aspect, as a time at which an instruction which uses data is *scheduled* for later execution is different from a time when an instruction which uses data is actually executed.

12. Claim 12 recites the limitation “a format predetermined to reduce a number of data streams for a level 3 linear algebra processing to be three streams” in lines 11-13. This limitation does not appear to be present in the original disclosure of the instant

application. If the limitation is present somewhere in one of the co-pending applications, this should be noted in any subsequent arguments to overcome the rejection. Applicant has previously argued that the present application supports the claim language of reducing the number of data streams to be three streams via various citations. However, the citations given do not appear to support the claim language of *reducing* the number of data streams to be three streams.

Examiner recommends rewording the claim to eliminate the “reducing” aspect, or giving a specific citation which explicitly notes that a reduction is occurring. It is noted that although a reduction may be occurring in view of past prior art implementations, the original disclosure nevertheless must disclose of this reduction in order for it to be valid inside the claim.

c. Claims 13-16 are rejected for inheriting the defects of base claim 12.

13. Claim 17 recites the limitation “instructions are inserted to move data into a cache providing data for said FPU before it is scheduled to be used in the linear algebra subroutine” in lines 6-7. The original disclosure does not disclose the broad interpretation of the claim in which the data is loaded into said Fregs before the instruction which uses said data is scheduled for execution. An amendment to overcome the corresponding indefinite rejection in a manner consistent with the original specification would most likely overcome this rejection as well.

Examiner recommends rewording the claim to eliminate the “scheduling” aspect, as a time at which an instruction which uses data is *scheduled* for later execution is different from a time when an instruction which uses data is actually executed.

14. Claim 17 recites the limitation “a format predetermined to reduce a number of data streams for a level 3 processing to be three streams” in lines 8-9. This limitation does not appear to be present in the original disclosure of the instant application. If the limitation is present somewhere in one of the co-pending applications, this should be noted in any subsequent arguments to overcome the rejection. Applicant has previously argued that the present application supports the claim language of reducing the number of data streams to be three streams via various citations. However, the citations given do not appear to support the claim language of *reducing* the number of data streams to be three streams.

Examiner recommends rewording the claim to eliminate the “reducing” aspect, or giving a specific citation which explicitly notes that a reduction is occurring. It is noted that although a reduction may be occurring in view of past prior art implementations, the original disclosure nevertheless must disclose of this reduction in order for it to be valid inside the claim.

d. Claims 18-19 are rejected for inheriting the defects of base claim 17.

15. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

16. Claims 1-9 and 11-19 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

17. Claim 1 recites the limitation "LSUs can load said data into said Fregs before it is scheduled to by used in said linear algebra subroutine execution" in lines 8-10. It is indefinite as to whether the data is loaded into said Fregs before the instruction which uses said data is executed, or whether the data is loaded into said Fregs before the instruction which uses said data is scheduled for execution, which occurs beforehand.

Examiner recommends rewording the claim to eliminate the "scheduling" aspect, as a time at which an instruction which uses data is *scheduled* for later execution is different from a time when an instruction which uses data is actually executed.

18. Claim 1 recites the limitation "p and q are small integers, meaning that p and q are sufficiently small" in lines 15-16. It is indefinite as to what are "small" integers or what "sufficiently small" integers are as whether an integer is small or not depends on what it is relative to. It is noted that the requirement that "pieces" of these blocks can be fitted into said FRegs would not appear to limit the sizes of the blocks in the context of the claim.

19. Claim 1 recites the limitation "the pieces of these blocks" in line 16. There is insufficient antecedent basis for this limitation in the claim.

20. Claim 1 recites the limitation "a level 3 nested loop matrix-matrix type kernel type operation processing (e.g., level 3 processing)" in lines 11-12. It is indefinite as to what "level 3 processing" refers to, as applicant appears to have given "level 3 processing" as

an example of a “level 3 nested loop matrix-matrix type kernel type operation processing”, but it is not known as to what specifically the level 3 processing is.

Examiner recommends removing (e.g., level 3 processing) and amending the other “level 3 processing” limitations to explicitly say “level 3 nested loop matrix-matrix type kernel type operation processing” to provide antecedent basis.

21. Claim 1 recites the limitation “in an optimal manner” in line 9. It is indefinite as to what constitutes an “optimal manner”, as any given method could be considered “optimal” in a wide variety of different metrics.

e. Claims 2-5 are rejected for failing to alleviate the rejection of claim 1 above.

22. Claim 6 recites the limitation “p and q are small integers” in line 15. It is indefinite as to what are “small” integers as whether an integer is small or not depends on what it is relative to. It is noted that the requirement that “pieces” of these blocks can be fitted into said FRegs would not appear to limit the sizes of the blocks in the context of the claim.

23. Claim 6 recites the limitation "the pieces of these blocks" in line 15-16. There is insufficient antecedent basis for this limitation in the claim.

24. Claim 6 recites the limitation “said three data streams comprise data of one matrix…and data for two remaining matrix operands…” in the last 4 lines. It is indefinite as to whether one data stream consists of only data of one matrix resident in said cache and the other two data streams each contains data for a respective remaining matrix

operand of the two matrix operands, or whether each data stream contains data of all three matrixes.

Examiner notes this issue has for the most part been corrected in the other independent claims.

25. Claim 6 recite the limitation “level 3 linear algebra processing” in line 11. It is indefinite as to what exactly a “level 3 linear algebra processing” is. Applicant argues that “Level 3 processing” is a commonly-used term by the DLA community to mean doing O(n^3) operations on O(n^2) data. However, page 12 of the instant specification discloses that the limitation “Level 3” means that the kernel involves three loops. Note that this definition does not necessarily mean that the loops are nested. Therefore, it is indefinite as to whether the aforementioned limitation means that the kernel involves three loops, or doing O(n^3) operations on O(n^2) data.

It is noted that this limitation had previously been corrected in, for example, claim 1, by reciting of a “level 3 nested loop matrix-matrix type kernel type operation”.

26. Claim 6 recites the limitation “K>1” in line 12. It is indefinite as to what the k variable is equivalent to.

27. Claim 6 recites the limitation “stride one (e.g., SIMD…K>1)” in line 12. It is indefinite as to how the concept of SIMD in general is an example of “stride one” as the general concept of SIMD does not by itself necessitate stride one data access.

f. Claims 7-9 and 11 are rejected for failing to alleviate the rejection of claim 6 above.

28. Claim 12 recites the limitation “inserting instructions to move data into said cache providing said data into said FPU before it was scheduled to be used for processing in said linear algebra subroutine” in line 8-10. It is indefinite as to whether the data is moved before the instruction which uses said data is executed, or whether the data is moved before the instruction which uses said data is scheduled for execution, which occurs beforehand. It is indefinite as to whether it is the moving of data into said cache or the providing data into said FPU which is done before it was scheduled to be used for processing in said linear algebra subroutine.

Examiner recommends rewording the claim to eliminate the “scheduling” aspect, as a time at which an instruction which uses data is *scheduled* for later execution is different from a time when an instruction which uses data is actually executed.

29. Claim 12 recites the limitation “p and q are small integers” in line 16. It is indefinite as to what are “small” integers as whether an integer is small or not depends on what it is relative to. It is indefinite as to what are “small” integers as whether an integer is small or not depends on what it is relative to. It is noted that the requirement that “pieces” of these blocks can be fitted into said FRegs would not appear to limit the sizes of the blocks in the context of the claim.

30. Claim 12 recites the limitation "the pieces of these blocks" in lines 16-17. There is insufficient antecedent basis for this limitation in the claim.

31. Claim 12 recite the limitation “level 3 linear algebra processing” in, for example, lines 12-13. It is indefinite as to what exactly a “level 3 linear algebra processing” is. Applicant argues that “Level 3 processing” is a commonly-used term by the DLA

community to mean doing O(n^3) operations on O(n^2) data. However, page 12 of the instant specification discloses that the limitation “Level 3” means that the kernel involves three loops. Note that this definition does not necessarily mean that the loops are nested. Therefore, it is indefinite as to whether the aforementioned limitation means that the kernel involves three loops, or doing O(n^3) operations on O(n^2) data.

It is noted that this limitation had previously been corrected in, for example, claim 1, by reciting of a “level 3 nested loop matrix-matrix type kernel type operation”.

32. Claim 12 recites the limitation “K>1 manner” in line 13. It is indefinite as to what the k variable is equivalent to.

33. Claim 12 recites the limitation “stride one (e.g., SIMD K>1 manner)” in line 13. It is indefinite as to how the concept of SIMD in general is an example of “stride one” as the general concept of SIMD does not by itself necessitate stride one data access.

g. Claims 13-16 are rejected for failing to alleviate the rejection of claim 12 above.

34. Claim 17 recites the limitation “instructions are inserted to move data into a cache providing data to said FPU before it is scheduled to be used in the linear algebra subroutine” in lines 6-7. It is indefinite as to whether the data is moved before the instruction which uses said data is executed, or whether the data is moved before the instruction which uses said data is scheduled for execution, which occurs beforehand. It is indefinite as to whether it is the moving of data into said cache or the providing data

into said FPU which is done before it was scheduled to be used for processing in said linear algebra subroutine.

Examiner recommends rewording the claim to eliminate the “scheduling” aspect, as a time at which an instruction which uses data is *scheduled* for later execution is different from a time when an instruction which uses data is actually executed.

35. Claim 17 recites the limitation “p and q are small integers” in line 12. It is indefinite as to what are “small” integers as whether an integer is small or not depends on what it is relative to. It is indefinite as to what are “small” integers as whether an integer is small or not depends on what it is relative to. It is noted that the requirement that “pieces” of these blocks can be fitted into said FRegs would not appear to limit the sizes of the blocks in the context of the claim.

36. Claim 17 recites the limitation "the pieces of these blocks" in lines 12-13. There is insufficient antecedent basis for this limitation in the claim.

37. Claim 17 recites the limitation “level 3 processing” in, for example, line 9 of claim 17. It is indefinite as to what exactly a “level 3 processing” is. Applicant argues that “Level 3 processing” is a commonly-used term by the DLA community to mean doing O(n^3) operations on O(n^2) data. However, it appears as though “level 3 processing” can also be interpreted as matrix-matrix operations. Although matrix-matrix operations may entail doing O(n^3) operations on O(n^2) data, it is readily recognized that there are other cases where O(n^3) operations are done on O(n^2) data that are unrelated to matrix-matrix operations. Therefore, it is indefinite as to whether Level 3 processing means doing O(n^3) operations on O(n^2) data or doing matrix-matrix operations, as

the former may be distinct from the latter. Moreover, page 12 of the instant specification discloses that the limitation “Level 3” means that the kernel involves three loops. Note that this definition does not necessarily mean that the loops are nested. Therefore, it is also indefinite as to whether the aforementioned limitation means that the kernel involves three loops, or doing O(n^3) operations on O(n^2) data.

It is noted that this limitation had previously been corrected in, for example, claim 1, by reciting of a “level 3 nested loop matrix-matrix type kernel type operation”.

38. Claim 17 recites the limitation “said three data streams comprise data of one matrix” in lines 14-17. The original disclosure does not disclose the broad interpretation of the claim in which each data stream contains data of all three matrixes. An amendment to overcome the corresponding indefinite rejection in a manner consistent with the original specification would most likely overcome this rejection as well.

Examiner notes that this rejection would be overcome if it is explicitly noted that one stream comprises the data resident in said cache, as done in independent claim 1.

39. Claim 17 recites the limitation “K>1 manner” in line 10. It is indefinite as to what the k variable is equivalent to.

40. Claim 17 recites the limitation “stride one (e.g., SIMD…K>1 manner)” in lines 9-10. It is indefinite as to how the concept of SIMD in general is an example of “stride one” as the general concept of SIMD does not by itself necessitate stride one data access.

h. Claims 18-19 are rejected for failing to alleviate the rejections of claim 17 above.

Claim Rejections - 35 USC § 102

41. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

42. Claims 1-9 and 11-19 are rejected under 35 U.S.C. 102(b) as being anticipated by Gustavson et al. (Gustavson) (Superscalar GEMM-based Level 3 BLAS – The On-going Evolution of a Portable and High-Performance Library, Para'98, pages 207-215).

43. Consider claim 1, Gustavson discloses for an execution code (section 1, line 6, BLAS code) controlling an operation of said floating point unit (FPU) (section 3.1, line 4, discloses floating point registers, therefore it is inherent there are floating point units that are doing the multiplications as in section 1, line 2) performing a linear algebra subroutine execution (section 1, line 8, routine along with section 1, line 1, linear algebra), inserting instructions to move data in a contiguous and stride one format (page 210, first indented paragraph, discloses of using regular load and store instruction to transfer data to and from registers; a load instruction loads contiguous data at an aligned memory address. Alternatively, section 4 describes of a PowerPC604 which performs loads to access data in a contiguous and stride one format) into a cache providing data for said FPU for direct loading in a stride one manner into said FPU (the L1 cache and registers are directly connected; see above regarding stride one format),

so that said LSUs can load said data into said Fregs in an optimal manner before it is scheduled to be used in said linear algebra subroutine execution (section 4.1, line 8, algorithmic prefetching), said data being prefetched into said cache from a memory in a register block format (the prefetching is described above in section 4.1, see below for the register block format explanations) to reduce a number of data streams for a level 3 nested loop matrix-matrix type kernel type operation processing (e.g. level 3 processing) to be three streams (section 3.1, first indented paragraph of page 210 as above, three total data streams are used, one for A, B, and C; note that as only a small square block of C instead of the entire C is being loaded into the registers, C is essentially a data stream of small square blocks. Also note that streams can be broadly read to be the data from the FPU registers to the FPU itself and thus encompasses A, B, and C regardless of the above technique) and to allow a loading of these streams into said FPU by said LSU (section 3.1, first indented paragraph of page 210 as above, number of load and store instructions), said register block format comprising a data storage format wherein data is stored in blocks of size p-by-q where p and q are small integers, meaning that p and q are sufficiently small so that the pieces of these blocks can be fitted into said Fregs (consider a subset or set of matrix data stored in any format in a memory. That matrix data can be arbitrarily split up into blocks of size p-by-q. Regardless of how small or big these blocks of matrix data are, and what data is within these blocks, single or multiple elements of this block of matrix data can be fitted in some way into said FRegs as is necessary for calculations to be subsequently performed), and wherein said three data streams comprise one stream of data of one

matrix of said level 3 processing is considered to be resident in said cache and one stream each for data for two remaining matrix operands of said level 3 processing as residing in a memory or a cache level higher than said cache (section 3.1, first indented paragraph of page 210 as above, three total data streams are used, one for A, B, and C; a small square block of C is being loaded into L0 cache, A and B reside in cache/memory).

44. Consider claim 6, Gustavson discloses an apparatus, comprising: a memory to store matrix data to be used for processing in a linear algebra program (section 4, line 12, shared main memory and section 4.2, lines 7-9, elements of the matrix); a floating point unit (FPU) to perform said processing (section 3.1, line 4, discloses floating point registers, therefore it is inherent there are floating point units that are doing the multiplications as in section 1, line 2); a load/store unit (LSU) to load data to be processed by said FPU (section 3.1, lines 6-7, load and store operations, thus it is inherent there is a load/store unit), said LSU loading said data into a plurality of floating point registers (FRegs) (section 3.1, line 4, floating point registers); and a cache to store data from said memory and provide said data to said Fregs (section 4.1, line 4, cache), wherein said matrix data in said memory is moved by having inserted moving instructions for said matrix data to be loaded into said cache prior to a need for said data to be loaded by said LSU into said Fregs for said processing, (section 4.1, line 8, algorithmic prefetching), said data being prefetched into said cache from said memory in a format (the prefetching is described above in section 4.1, see below for the register

block format explanations) predetermined to reduce a number of data streams for a level 3 processing to be three streams (section 3.1, first indented paragraph of page 210 as above, three total data streams are used, one for A, B, and C; note that as only a small square block of C instead of the entire C is being loaded into the registers, C is essentially a data stream of small square blocks. Also note that streams can be broadly read to be the data from the FPU registers to the FPU itself and thus encompasses A, B, and C regardless of the above technique) and to allow a stride one (e.g., SIMD (single instruction, multiple data) k>1) loading of these streams into said FPU by said LSU (see the second-to-last paragraph of section 3.1, multiple element load instructions; alternatively, page 210, first indented paragraph, discloses of using regular load and store instruction to transfer data to and from registers; a load instruction loads contiguous data at an aligned memory address. Alternatively, section 4 describes of a PowerPC604 which performs loads to access data in a contiguous and stride one format; note that the SIMD K>1 portion of the claim does not have to be given patentable weight due to its inclusion as an example), wherein said format comprises a register block format wherein data is stored in blocks of size p-by-q where p and q are small integers so that the pieces of these blocks can be fitted into said Fregs (consider a subset or set of matrix data stored in any format in a memory. That matrix data can be arbitrarily split up into blocks of size p-by-q. Regardless of how small or big these blocks of matrix data are, and what data is within these blocks, single or multiple elements of this block of matrix data can be fitted in some way into said FRegs as is necessary for calculations to be subsequently performed), and wherein said three data

streams comprise data of one matrix of said level 3 linear algebra processing is considered to be resident in said cache and two remaining matrix operands of said level 3 linear algebra processing reside in a memory or a cache level higher than said cache (section 3.1, first indented paragraph of page 210 as above, three total data streams are used, one for A, B, and C; a small square block of C is being loaded into L0 cache, A and B reside in cache/memory).

45. Consider claim 12, Gustavson discloses for an execution code (section 1, line 6, BLAS code) controlling an operation of said floating point unit (FPU) (section 3.1, line 4, discloses floating point registers, therefore it is inherent there are floating point units that are doing the multiplications as in section 1, line 2) performing a linear algebra subroutine execution (section 1, line 8, routine along with section 1, line 1, linear algebra), inserting instructions to move data into said cache providing data into said FPU before it was scheduled to be used for processing in said linear algebra subroutine (section 4.1, line 8, algorithmic prefetching), wherein said data is prefetched into said cache from a memory in a format (the prefetching is described above in section 4.1, see below for the register block format explanations) to reduce a number of data streams for a level 3 linear algebra processing to be three streams (section 3.1, first indented paragraph of page 210 as above, three total data streams are used, one for A, B, and C; note that as only a small square block of C instead of the entire C is being loaded into the registers, C is essentially a data stream of small square blocks. Also note that streams can be broadly read to be the data from the FPU registers to the FPU itself and

thus encompasses A, B, and C regardless of the above technique) and to allow a stride one (e.g., SIMD k>1 manner) loading of these streams into said FPU by said LSUs (page 210, first indented paragraph, discloses of using regular load and store instruction to transfer data to and from registers; a load instruction loads contiguous data at an aligned memory address. Alternatively, section 4 describes of a PowerPC604 which performs loads to access data in a contiguous and stride one format. Also, see the second-to-last paragraph of section 3.1, multiple element load instructions. Alternatively, section 4 describes of a PowerPC604 which performs loads to access data in a contiguous and stride one format; note that the SIMD K>1 portion of the claim does not have to be given patentable weight due to its inclusion as an example), wherein said nonstasndard format comprises a register block format wherein data is stored in blocks of size p-by-q where p and q are small integers so that the pieces of these blocks can be fitted into said Fregs (consider a subset or set of matrix data stored in any format in a memory. That matrix data can be arbitrarily split up into blocks of size p-by-q. Regardless of how small or big these blocks of matrix data are, and what data is within these blocks, single or multiple elements of this block of matrix data can be fitted in some way into said FRegs as is necessary for calculations to be subsequently performed), and wherein said three data streams comprise data of one matrix of said level 3 linear algebra processing is considered to be resident in said cache and data for two remaining matrix operands of said level 3 linear algebra processing reside in a memory or a cache level higher than said cache (section 3.1, first indented paragraph of

page 210 as above, three total data streams are used, one for A, B, and C; a small square block of C is being loaded into L0 cache, A and B reside in cache/memory).

46. Consider claim 17, Gustavson discloses a method of providing a service involving at least one of solving and applying a scientific/engineering problem, said method comprising at least one of:

using a linear algebra software package that computes one or more matrix subroutines, wherein said linear algebra software package generates an execution code (section 1, line 6, BLAS code) controlling an operation of a floating point unit (FPU) (section 3.1, line 4, discloses floating point registers, therefore it is inherent there are floating point units that are doing the multiplications as in section 1, line 2) performing a linear algebra subroutine execution (section 1, line 8, routine along with section 1, line 1, linear algebra), such that instructions are inserted to move data into a cache providing data for said FPU before it is scheduled to be used in the linear algebra subroutine (section 4.1, line 8, algorithmic prefetching), said data being prefetched from a memory in a format (the prefetching is described above in section 4.1, see below for the register block format explanations) to reduce a number of data streams for a level 3 processing to be three streams (section 3.1, first indented paragraph of page 210 as above, three total data streams are used, one for A, B, and C; note that as only a small square block of C instead of the entire C is being loaded into the registers, C is essentially a data stream of small square blocks. Also note that streams can be broadly

read to be the data from the FPU registers to the FPU itself and thus encompasses A, B, and C regardless of the above technique) and to permit a stride one (e.g., SIMD (single instruction, multiple data) k>1 loading of these streams into said FPU (page 210, first indented paragraph, discloses of using regular load and store instruction to transfer data to and from registers; a load instruction loads contiguous data at an aligned memory address. Alternatively, section 4 describes of a PowerPC604 which performs loads to access data in a contiguous and stride one format; also, see the second-to-last paragraph of section 3.1, multiple element load instructions. Alternatively, section 4 describes of a PowerPC604 which performs loads to access data in a contiguous and stride one format; note that the SIMD K>1 portion of the claim does not have to be given patentable weight due to its inclusion as an example), wherein said format comprises a register block format wherein data is stored in blocks of size p-by-q where p and q are small integers so that the pieces of these blocks can be fitted into said Fregs (consider a subset or set of matrix data stored in any format in a memory. That matrix data can be arbitrarily split up into blocks of size p-by-q. Regardless of how small or big these blocks of matrix data are, and what data is within these blocks, single or multiple elements of this block of matrix data can be fitted in some way into said FRegs as is necessary for calculations to be subsequently performed), and wherein said three data streams comprise data of one matrix of said level 3 linear algebra processing that reside in said cache and one stream each for two remaining matrix operands of said level 3 linear algebra processing reside in a memory or a cache level higher than said cache (section 3.1, first indented paragraph of page 210 as above, three total data streams are

used, one for A, B, and C; a small square block of C is being loaded into L0 cache, A and B reside in cache/memory),

providing a consultation for solving a scientific/engineering problem using said linear algebra software package (it is inherent that the BLAS will solve some type of scientific/engineering problem for someone who may or may not be the operator of the BLAS program); transmitting a result of said linear algebra software package on at least one of a network, a signal-bearing medium containing machine-readable data representing said result, and a printed version representing said result; and receiving a result of said linear algebra software package on at least one of a network, a signal-bearing medium containing machine-readable data representing said result, and a printed version representing said result (it is inherent that the result of the problem will be conveyed to someone who may or may not be the operator of the BLAS program; furthermore, it is inherent that the result can only be shown either through a printout or through some type of electronic means, which encompasses voice through a phone or data through a network that is read via a monitor).

47. Consider claims 2, 11, and 13, Gustavson discloses said moving data is accomplished by scheduling move type instructions into time slots existing in a Level 3 Dense Linear Algebra Subroutine. As explained above, it is inherent to prefetching that data is loaded into the cache before the instruction that needs that data is executed, thus there must be a difference between the time of that instruction execution and the time of its data loading, otherwise it would not be prefetching. Furthermore, Gustavson

discloses in page 12, lines 2-3 of section 4.1 that the prefetching instruction does not disturb ongoing computations and data references, thus this prefetching must be done in “time slots” which are independent of other instruction fetching. Gustavson in section 3, line 5, discloses of DGEMM, which is a type of Level 3 Dense Linear Algebra Subroutine.

48. Consider claims 3, 7, and 14, Gustavson discloses said linear algebra subroutine comprises a matrix multiplication operation (section 1, line 2, matrix multiply).

49. Consider claims 4, 8, 15, and 18, Gustavson discloses said matrix subroutine comprises an equivalent of a subroutine from a LAPACK (Linear Algebra PACKage) (section 1, line 1, discloses a BLAS, which is a part of LAPACK).

50. Consider claims 5, 9, 16, and 19, Gustavson discloses said linear algebra subroutine comprises a BLAS Level 3 L1 cache kernel (Abstract, lines 1-6, level 3 BLAS kernel and level 1 cache).

Response to Arguments

51. Examiner first notes that due to the claim amendments made in the copending application 10671937, examiner is withdrawing the previously made provisional double patenting rejection.

52. Applicant argues that the claim amendments overcome the previously made 112 issues. However, a large portion of the previously made 112 rejections have been neither specifically addressed nor had their respective claim limitations amended in any way. These 112 rejections are thus maintained.

53. Applicant argues on page 15 that the prior art reference used in the rejection above refers only to multiple loads of load multiple type k=1, whereas the present application addresses architecture capable of a SIMD load with k>1.

However, the addition of the limitation “k>1” into the instant set of claims is indefinite and, as a consequence of this indefiniteness, the prior rejections remains valid. It is indefinite as to what “k” is referring to, and thus k can be broadly read to be the amount of data which is loaded. Examiner recommends explicitly reciting the architectural details which are indicative of “k>1”. As examiner noted in the previous interview, definite language must be used in order to overcome the prior art. As also explained in the 112 rejection, the inclusion of “k>1” within the context of the e.g. clause may also prevent the limitation from being given patentable weight. Examiner also notes that the entire clause "to allow a stride one (e.g., SIMD (single instruction, multiple data) k>1) loading of these streams into said FPU by said LSU" is an intended use limitation and may also prevent the limitation from being given patentable weight.

54. Applicant also argues on page 15 that the citation used to teach the limitation “format” have no suggestion whatsoever as to whether the data is a specific format.

However, it is necessarily the case that the data is in a data format in general, even if it is a “standard” format. In other words, it is readily recognized to one of ordinary skill in the art that, for the technique in the prior art to work, the matrix data has to be formatted in memory in a specific manner and not arbitrarily. This format, whether standard or non-standard, still meets the claimed limitations which elaborate on the format, namely the limitations regarding the register block format.

55. Examiner recommends first amending the claims so that all of the above 112 issues are addressed and overcome, and then amending the claims (and adding supporting disclosure to the specification from the co-pending applications if necessary) to either disclose the specifics of the register block format, or elaborating on the SIMD aspect of the invention in a definite way such that an art which discloses of SIMD operation of matrix data in a standard format does not read on the claimed limitations.

Conclusion

56. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

57. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Keith Vicary whose telephone number is (571)270-1314. The examiner can normally be reached on Monday - Thursday, 6:15 a.m. - 5:45 p.m., EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on 571-272-4162. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Eddie P Chan/
Supervisory Patent Examiner, Art Unit 2183

/Keith Vicary/
Examiner, Art Unit 2183